Modèle de régression multi-tâche par processus gaussiens avec moyenne informée

Arthur LEROY[*]
Servane GEY[†]
Pierre LATOUCHE[‡]
Benjamin GUEDJ[§]

Résumé : La régression par processus gaussien est un outil classique de l'apprentissage supervisé qui présente l'avantage de fournir un cadre probabiliste permettant une quantification de l'incertitude des prédictions. L'apprentissage dans de tels modèles se concentre généralement sur l'estimation des hyper-paramètres du noyau associé à la structure de covariance, et non sur la moyenne a priori du processus. Ainsi, la qualité de prédiction peut se retrouver fortement affectée dès lors que l'on s'éloigne des points d'observation avec une moyenne a priori non pertinente. Ce travail propose l'utilisation d'un modèle multi-tâche, dont les données de plusieurs individus sont supposées partager une structure commune. Cette structure est modélisée par un processus moyen, commun à tous les individus, permettant une prediction plus pertinente même lorsque les observations d'un nouvel individu sont peu nombreuses ou mal reparties sur l'espace des entrées. L'apprentissage des hyper-paramètres du modèle est effectué par un algorithme EM parallélisable. Une étape de calcul de la loi a posteriori du processus moyen est ensuite effectuée pour intégrer l'information de tous les individus des données d'apprentissage. Enfin, après marginalisation, l'étape de prédiction est analogue au cas classique de la régression par processus gaussien. Les prédictions ainsi obtenues pour un nouvel individu sont centrées sur des valeurs informées par les autres individus, avec prise en compte des deux sources d'incertitude distinctes. Une étude de simulations illustre les bénéfices et les coûts calculatoires d'une telle approche.

Mots clefs. Processus Gaussiens, Apprentissage multi-tâche, Algorithme EM, ...

Abstract. Gaussian process regression is a common tool of supervised learning that provides a convenient probabilistic framework, leading to predictions associated to uncertainty quantification. The learning step in such a model generaly focuses on hyper-parameters estimation of the kernel associated to the covariance structure, rather than the prior mean of the process. Therefore, the quality of prediction might severely decrease, with an unappropriate prior mean, as we move away from observation points. This work presents a multi-task model, where data come from several individuals supposed to share some structure altogether. We model this structure through a mean process, common to all individuals, leading to more reliable predictions even though a new individual is observed on few or sparse input locations. An EM algorithm, that can be parallelized, is used to learn the model hyper-parameters. An additional step integrates knowledge from all individuals in the training dataset through the computation of the mean process posterior. Finally, after marginalization, the prediction step is analogous to the classic gaussian process regression. Such predictions for a new individual are centered on values informed by the other individuals, with uncertainty coming from the two distinct sources. A simulation study illustrates the advantages and computational costs of such an approach.

Keywords. Gaussian Processes, Multi-task learning, EM algorithm, ...

Contexte

L'apprentissage par processus gaussiens (GP, pour 'Gaussian Processes' en anglais) a fait l'objet de nombreuses études durant ces deux dernières décennies, dont l'ouvrage de référence sur le sujet est Rasmussen and Williams (2006). Cette approche non-paramétrique permet d'estimer une fonction d'apprentissage dans

[*]MAP5 - Université de Paris, arthur.leroy@parisdescartes.fr
[†]MAP5 - Université de Paris, servane.gey@parisdescartes.fr
[‡]MAP5 - Université de Paris, pierre.latouche@parisdescartes.fr
[§]MODAL - INRIA / University College London, benjamin.guedj@inria.fr

un cadre probabiliste en posant une hypothèse a priori sur la forme de cette fonction. La simplicité de la méthode, ainsi que la possibilité d'obtenir des intervalles de crédibilité pour les prédictions, ont participé à son succès. Néanmoins, son coût calculatoire en $\mathcal{O}(N^3)$ en complique l'utilisation pour des grands jeux de données.

Une littérature très riche se concentre sur la problématique de donner de bonnes approximations lorsque le nombre N d'observations d'un processus est trop grand pour permettre l'inférence exacte. Dans le cas de l'étude de plusieurs tâches, ou individus, ayant chacun un nombre d'observations raisonnable, une littérature existe au croisement entre l'apprentissage multi-tâche et les GPs.

Un des premiers papiers, Yu, Tresp, and Schwaighofer (2005), qui développe un modèle bayésien hiérarchique, définit des paramètres de moyenne et de covariance communs entre les individus et estimés par un algorithme EM. Ensuite, Bonilla, Chai, and Williams (2008) se concentrent sur le partage de la structure de covariance entre les différents individus, alors que dans J. Q. Shi and Wang (2008), les auteurs proposent un premier modèle dans lequel il existe une fonction moyenne commune, à estimer, pour chaque GP individuel. Cette approche fait l'objet de plusieurs articles, ainsi que d'un package R associé, Jian Qing Shi and Cheng (2014). Pour estimer cette fonction moyenne, les auteurs proposent d'utiliser une décomposition dans une base de B-splines. Cela constitue donc une approche paramétrique déterministe, ignorant l'incertitude liée à l'estimation de cette fonction moyenne. L'objectif de notre travail réside donc dans la définition d'une extension de cette idée, en modélisation la moyenne a priori également par un GP, qui sera alors estimé à l'aide de tous les individus. Cette approche présente l'avantage d'offrir un cadre non-paramétrique global et une prise en compte de l'incertitude liée au processus moyen commun. Le coût calculatoire à payer en contrepartie reste raisonnablement du même ordre tant que l'union des temps d'observations (les entrées sont souvent assimilées au temps dans la littérature) de tous les individus ne grandit pas outre mesure.

A noter que l'article Yang et al. (2016) présente des idées du même ordre dans un cadre un peu différent, en définissant un modèle hiérarchique et un algorithme MCMC associé. Le coût calculatoire de cette procédure pouvant être rapidement trop important, nous tentons également dans notre approche d'utiliser les propriétés agréables des GPs pour calculer exactement les log-vraisemblances utilisées dans l'apprentissage, qui est effectué à l'aide d'un algorithme EM.

Quelques notations

- M le nombre d'individus,
- N_i le nombre d'observations pour l'individu i,

On dispose d'un échantillon $\{(\mathbf{y}_i, \mathbf{t}_i)\}_{i=1,\dots,M}$, tel que:

- $\mathbf{t}_i = (t^1, \dots, t^{N_i})^T$ le vecteur des temps de l'individu i,
- $\mathbf{y}_i = y_i(\mathbf{t}_i) = (y_i(t^1), \dots, y_i(t^{N_i}))^T$ le vecteur des observations de l'individu i,
- $\mathbf{t} = \bigcup_i \mathbf{t}_i$ l'union de tous les points de temps observés,
- $N = card(\mathbf{t})$, le nombre total de points de temps distincts,
- K_{θ_0} un noyau d'hyper-paramètres θ_0,
- m_0 une fonction réelle donnée comme moyenne a priori pour le processus moyen, noté μ_0,
- $(\Sigma_{\theta_i})_i$ un ensemble de noyaux de même forme et d'hyper-paramètres respectifs $(\theta_i)_i$,
- $\sigma_i^2 \in \mathbb{R}$, $\forall i$,
- $\Theta = \{\theta_0, (\theta_i)_i, (\sigma_i^2)_i\}$ le vecteur des hyper-paramètres du modèle.

Modèle et hypothèses

On pose le modèle suivant:

$$\forall i, \forall t, \quad y_i(t) = \mu_0(t) + f_i(t) + \epsilon_i(t),$$

- $\mu_0(\cdot) \sim GP(m_0(\cdot), K_{\theta_0}(\cdot, \cdot))$,

- $f_i(\cdot) \sim GP(0, \Sigma_{\theta_i}(\cdot,\cdot))$, $(f_i)_i \perp\!\!\!\perp$,
- $\epsilon_i(t) \sim \mathcal{N}(0, \sigma_i^2)$, $(\epsilon_i)_i \perp\!\!\!\perp, \forall t \in \mathbb{R}$,
- $\mu_0 \perp\!\!\!\perp (f_i)_i$.

On note $\Psi_i(\cdot,\cdot) = \Sigma_{\theta_i}(\cdot,\cdot) + \sigma_i^2 I_d$, et on en déduit:

$$y_i(\cdot)|\mu_0 \sim GP(\mu_0(\cdot), \Psi_i(\cdot,\cdot)), \quad (y_i|\mu_0)_i \perp\!\!\!\perp$$

Si on applique cela à notre échantillon, on a la loi a priori suivante:

$$y_i(\mathbf{t}_i)|\mu_0(\mathbf{t}_i) \sim \mathcal{N}(\mu_0(\mathbf{t}_i), \Psi_i(\mathbf{t}_i, \mathbf{t}_i)),$$

avec

$$\Psi_i(t_k, t_l) = cov(y_i(t_k), y_i(t_l)), \quad \forall t_k, t_l \in \mathbf{t}_i.$$

L'apprentissage

Dans un modèle de régression GP, il est généralement nécessaire d'apprendre un nombre limité d'hyper-paramètres, qui caractérisent le noyau associé à la fonction de covariance. Dans notre cadre, puisque les observations sont supposées être la somme de deux GPs indépendants, il faut également apprendre les hyper-paramètres θ_0 du noyau K_{θ_0} du processus moyen. De plus, nous verrons dans l'étape de prédiction que ce processus moyen est estimé grâce aux observations issues de nos échantillons d'apprentissage. Il est donc aussi nécessaire d'apprendre les (θ_i, σ_i^2), pour tout $i = 1, \dots, M$. Il est important de noter que le processus moyen μ_0 est par définition commun à tous les individus, et son estimation est dépendante des hyper-paramètres Θ. Une procédure classique dans ce contexte est l'utilisation d'un algorithme EM, qui procède en alternance au calcul de la loi de μ_0 avec Θ fixé, puis à l'estimation des hyper-paramètres par maximisation de log-vraisemblance (qui fait intervenir $p(\mu_0)$). Généralement, un tel algorithme itératif converge vers des maxima locaux en peu d'itérations, et différents choix d'initialisations peuvent aider à trouver un maximum global.

Etape E:

$$\begin{aligned} p(\mu_0(\mathbf{t})|(\mathbf{y}_i)_i, \Theta) &\propto \overbrace{p((\mathbf{y}_i)_i|\mu_0(\mathbf{t}), (\sigma_i^2)_i, (\theta_{\theta_i})_i))}^{\prod_{i=1}^{M} \mathcal{N}(\mu_0(\mathbf{t}), \Psi_i)} \overbrace{p(\mu_0(\mathbf{t})|\theta_0)}^{\mathcal{N}(m_0, K_{\theta_0})} \\ &= \mathcal{N}(\hat{m}_0(\mathbf{t}), \hat{K}), \end{aligned}$$

avec:

- $\hat{K} = \left((K_{\theta_0})^{-1} + \sum_{i=1}^{M} (\Psi_i)^{-1} \right)^{-1}$,
- $\hat{m}_0(\mathbf{t}) = \hat{K} \left(K_{\theta_0}^{-1} m_0 + \sum_{i=1}^{M} (\Psi_i)^{-1} \mathbf{y}_i \right)$.

NB: les matrices $(\Psi_i)_i$ et vecteurs $\boldsymbol{y}_i$ sont en fait ici complétés avec des 0 pour être de dimension N, tout comme $\mu_0(\boldsymbol{t})$. Il s'agit de détails techniques que nous omettons par soucis de concision.

Etape M:

$$\hat{\Theta} = arg \max_{\Theta} \mathbb{E}_{\mu_0} \left[log\ p(\mu_0(\mathbf{t}), (\mathbf{y}_i)_i, \Theta) \right]$$

Il est intéressant de noter que cette maximisation, relativement compliquée à première vue, peut se découper assez simplement en $M+1$ maximisations par indépendance des individus entre eux. Ce qui a pour avantage de devoir optimiser à chaque fois une fonction avec un faible nombre de paramètres, et de pouvoir effectuer les calculs indépendamment en parrallèle les uns des autres.

La prédiction

L'objectif réside dans la prédiction de la quantité $p(y_*(t)|y_*(\mathbf{t}_*), (\mathbf{y}_i)_i)$, définissant la loi d'un nouvel individu d'indice $*$ en un temps non observé t, en connaissant les observations de ce nouvel individu et les observations de tous les autres individus.

Calcul de la loi a posteriori de μ_0

Une fois les hyper-paramètres du modèle appris, il est nécessaire d'effectuer le calcul de la loi à posteriori du processus moyen μ_0, pour les temps auxquels on souhaite obtenir une prédiction. Il s'agit d'une étape spécifique à notre procédure, qui n'existe pas dans une régression GP classique, et qui permet d'obtenir une moyenne informée pour notre prédiction finale.

Par concision, on note $\mathbf{t}_+ = (t, \mathbf{t}_*)$ le vecteur de tous les temps observés, auquel on ajoute le temps à prédire.

$$\begin{aligned} p(y_*(t), y_*(\mathbf{t}_*)|(\mathbf{y}_i)_i) &= \int p(y_*(t), y_*(\mathbf{t}_*), \mu_0(\mathbf{t}_+)|(\mathbf{y}_i)_i)\ d\mu_0(\mathbf{t}_+) \\ &= \int p(y_*(t), y_*(\mathbf{t}_*)|(\mathbf{y}_i)_i, \mu_0(\mathbf{t}_+))\ p(\mu_0(\mathbf{t}_+)|(\mathbf{y}_i)_i)\ d\mu_0(\mathbf{t}_+) \\ &\underbrace{=}_{(\mathbf{y}_i)_i | \mu_0 \perp\!\!\!\perp} \int p(y_*(t), y_*(\mathbf{t}_*)|\mu_0(\mathbf{t}_+))\ p(\mu_0(\mathbf{t}_+)|(\mathbf{y}_i)_i)\ d\mu_0(\mathbf{t}_+). \end{aligned}$$

Or, par définition du modèle, on a:

$$p(y_*(t), y_*(\mathbf{t}_*)|\mu_0(\mathbf{t}_+)) = \mathcal{N}\left(\begin{bmatrix}\mu_0(t)\\ \mu_0(\mathbf{t}_*)\end{bmatrix}, \begin{pmatrix}\Psi_*(t,t) & \Psi_*(t,\mathbf{t}_*)\\ \Psi_*(\mathbf{t}_*,t) & \Psi_*(\mathbf{t}_*,\mathbf{t}_*)\end{pmatrix}\right).$$

De plus, pendant l'étape E de l'entrainement, on a vu que:

$$p(\mu_0(\mathbf{t}_+)|(\mathbf{y}_i)_i) = \mathcal{N}(\hat{m}_0(\mathbf{t}_+), \hat{K}(\mathbf{t}_+,\mathbf{t}_+)) = \mathcal{N}\left(\begin{bmatrix}\hat{m}_0(t)\\ \hat{m}_0(\mathbf{t}_*)\end{bmatrix}, \begin{pmatrix}\hat{K}(t,t) & \hat{K}(t,\mathbf{t}_*)\\ \hat{K}(\mathbf{t}_*,t) & \hat{K}(\mathbf{t}_*,\mathbf{t}_*)\end{pmatrix}\right).$$

Avec $\hat{m}_0$ et $\hat{K}$ définis comme dans l'étape E, on a finalement:

$$p(y_*(t), y_*(\mathbf{t}_*)|(\mathbf{y}_i)_i) = \mathcal{N}(\hat{m}_0(\mathbf{t}_+), \Gamma_*(\mathbf{t}_+,\mathbf{t}_+)) = \mathcal{N}\left(\begin{bmatrix}\hat{m}_0(t)\\ \hat{m}_0(\mathbf{t}_*)\end{bmatrix}, \begin{pmatrix}\Gamma_*(t,t) & \Gamma_*(t,\mathbf{t}_*)\\ \Gamma_*(\mathbf{t}_*,t) & \Gamma_*(\mathbf{t}_*,\mathbf{t}_*)\end{pmatrix}\right),$$

avec $\Gamma_* = \Psi_* + \hat{K}$.

La prédiction

En utilisant la formule habituelle de prédiction GP, on obtient la loi a posteriori suivante:

$$p(y_*(t)|y_*(t_*), (\mathbf{y}_i)_i) = \mathcal{N}(m_*, v_*),$$

avec :

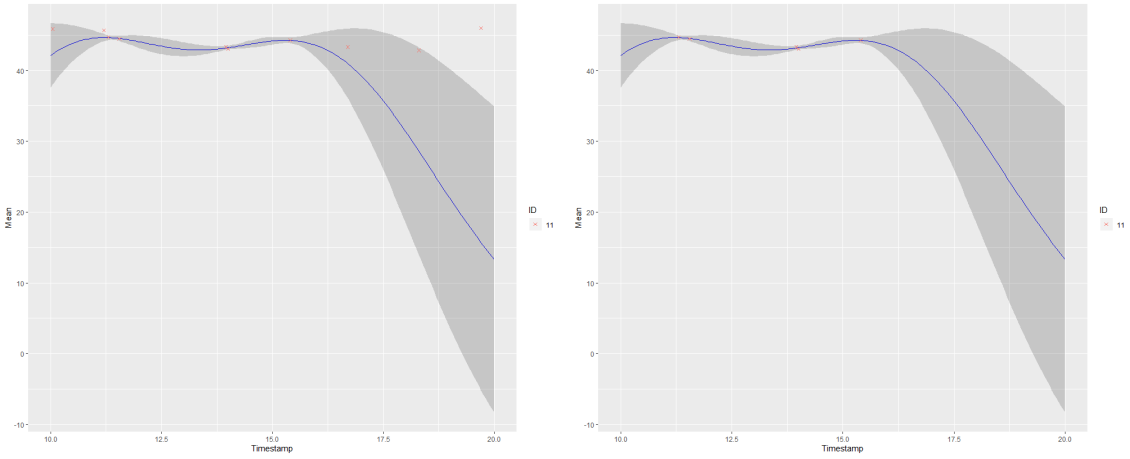
- $m_* = \hat{m}_0(t) + \Gamma_*(t, \mathbf{t}_*)\Gamma_*(\mathbf{t}_*, \mathbf{t}_*)^{-1}(y_*(t_*) - \hat{m}_0(\mathbf{t}_*))$,
- $v_* = \Gamma_*(t, t) - \Gamma_*(t, \mathbf{t}_*)\Gamma_*(\mathbf{t}_*, \mathbf{t}_*)^{-1}\Gamma_*(\mathbf{t}_*, t)$.

Visualisation graphique

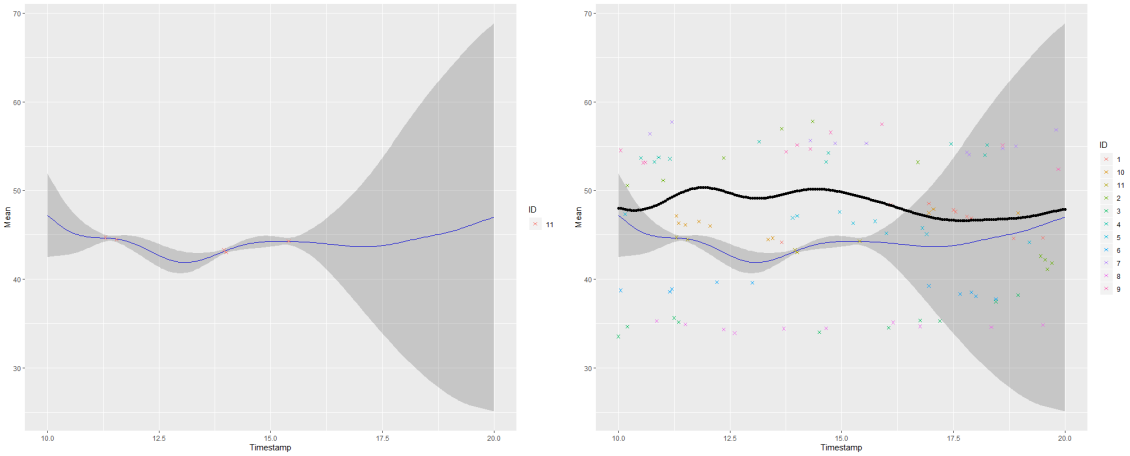
Une étude de simulation a été effectuée pour comparer une approche classique de modélisation par un unique GP avec la procédure décrite précédemment. Des données ont été simulées pour 11 individus, en tirant pour 10 temps d'observation et des hyper-paramètres spécifiques à chacun, des observations issues de GPs de même fonction moyenne (constante égale à 45) et de même forme de fonction de covariance (noyau exponentiel).

Le processus moyen μ_0 de notre procédure a été estimé à l'aide des 10 premiers individus et la prédiction s'effectue sur le 11ème individu, pour lequel on observe seulement 4 temps tirés aléatoirement. Dans les deux cas, la moyenne a priori (m_0) a été fixée à 0, afin de simuler l'absence d'information préalable. Sur les graphiques ci-dessous, on peut comparer les résultats de cette prediction sur une grille de 200 points de temps, pour chacune des deux approches.

Pour des raisons de concision, nous omettons ici les résultats plus détaillés concernant l'évaluation des erreurs, la comparaison avec d'autres algorithmes, les temps de calculs et la performance sur des cas pathologiques. Ces détails seront évoqués lors de la présentation orale.



La figure ci-dessus montre le résultat de la prédiction avec un modèle GP classique, qui a été entrainé seulement 4 observations représentées ici par les croix rouges de la figure de droite. Sur la figure de gauche, les croix rouges représentent la totalité des observations simulées. On voit que la prédiction reste fiable lorsque l'on reste proche des données observées, mais plongent rapidement vers la valeur a priori 0 dès que l'on s'en éloigne.



Les figures ci-dessus montrent le résultat de la prédiction avec le modèle décrit précédemment, qui a été entrainé sur la totalité des observations des 10 premiers individus, et sur les 4 mêmes observations du 11ème individu. La figure de gauche illustre cette prédiction dans le même contexte que pour un seul GP, mais qui cette fois reste bien meilleur, même lorsque l'on s'éloigne des points d'observation. Ceci s'explique à l'aide du graphique de droite, qui montre la totalité des observations utilisées (en couleur), et la valeur du processus moyen (points noirs) pour tous les temps de la grille de prédiction. En effet, lorsque l'on s'éloigne des temps observation, la prédiction se rapproche de la moyenne a priori, qui est cette fois informée par les autres individus.

Travaux en cours

Dans le cas où les courbes observées sont supposées issues de GPs avec des processus moyens différents, il est également possible de poser un modèle qui intègre une partie *clustering* au sein de la méthode décrite précédemment. Dans ce contexte, on suppose qu'il existe un nombre k de clusters dont chacun est caracterisé par un processus moyen μ_k spécifique. Il est alors possible d'écrire une procédure d'apprentissage qui ne sera plus un EM classique, mais un EM variationnel (une approche déjà utilisée dans le contexte GP dans Titsias (2009)) compte tenu des relations de dépendance entre les variables latentes. La prédiction quant à elle se fait de manière relativement naturelle par la suite. Ce travail est actuellement en cours d'implémentation.

Références

Bonilla, Edwin V, Kian M. Chai, and Christopher Williams. 2008. "Multi-Task Gaussian Process Prediction." In *Advances in Neural Information Processing Systems 20*, edited by J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, 153–60. Curran Associates, Inc.

Rasmussen, Carl Edward, and Christopher K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. Cambridge, Mass: MIT Press.

Shi, J. Q., and B. Wang. 2008. "Curve Prediction and Clustering with Mixtures of Gaussian Process Functional Regression Models." *Statistics and Computing* 18 (3): 267–83. doi:10.1007/s11222-008-9055-1.

Shi, Jian Qing, and Yafeng Cheng. 2014. "Gaussian Process Function Data Analysis R Package 'GPFDA'," 33.

Titsias, Michalis K. 2009. "Variational Learning of Inducing Variables in Sparse Gaussian Processes." *AISTATS*, 8.

Yang, Jingjing, Hongxiao Zhu, Taeryon Choi, and Dennis D. Cox. 2016. "Smoothing and MeanCovariance Estimation of Functional Data with a Bayesian Hierarchical Model." *Bayesian Analysis* 11 (3): 649–70. doi:10.1214/15-BA967.

Yu, Kai, Volker Tresp, and Anton Schwaighofer. 2005. "Learning Gaussian Processes from Multiple Tasks." In *Proceedings of the 22Nd International Conference on Machine Learning*, 1012–9. ICML '05. Bonn, Germany: ACM. doi:10.1145/1102351.1102479.